A Review of Wavelet-based Conjugate Gradient Method for Solving Poisson Equations

Nobuatsu Tanaka *

ABSTRACT

This is a review of the paper [10] that describes a powerful and simple new wavelet-based preconditioning method for solving large systems of linear equations and shows how it is implemented in simulations of fluid flow modeled by Poisson equations. The linear systems can be solved with an iterative matrix solver, however, in the absence of our method, the computing time will increase markedly with respect to an increase in grid points. Use of our technique leads to a matrix with a bounded condition number so that computing time is reduced significantly. Our technique also overcomes difficulties associated with earlier wavelet-based solvers, which assume periodic boundary conditions making them difficult to apply to real simulation problems. Results from our numerical experiments with one- and two-dimensional Poisson equations confirm the power and accuracy of our wavelet-based preconditioning method. Unlike many preconditioning methods which are not suitable for vector and parallel processing, our algorithm can take advantage of the extra processing capabilities and enhance computing performance. For example, a speed up of over 100 fold can be achieved when solving Poisson equations on a Cray T3D using 128 processors in parallel.

Key Words: wavelet, preconditioning, Poisson equation, matrix solver, conjugate gradient method

1 Introduction

Numerical simulations play an important role in the analysis of many kinds of complex phenomena which occur in nuclear power plants, such as neutron transport, fluid flow and structural behavior. The need for and use of detailed and accurate numerical simulations has been increasing in many engineering fields since the cost of performing physical experiments is considerably higher. This trend is also fueled by the development of more powerful computer hardware, such as vector supercomputers and massively parallel processors. However, physical experiments cannot be eliminated altogether; we must continue to conduct them to some extent to understand physical phenomena and to assist in drawing correlations with simulation models and computer codes and to obtain data for the qualification of these codes.

Scientists need simple mechanistic models with few assumptions which yield reliable and accurate results in numerical simulations. Sophisticated models are often not appropriate or possible in the fluid flow analysis of nuclear power plants. For example, when the Direct Numerical Simulation method (one of the most detailed models of fluid flow) is used to simulate nuclear power plant conditions with high Reynolds number, turbulent flow, and systems with complex geometries, a very fine mesh is needed, and computing time increases drastically with an increase in mesh points. The most time-consuming step of the calculation is solving continuous equations governing phenomena. To solve these equations, scientists first approximate them by discretization, then transform them into a large system of linear equations.

* Nuclear Engineering Laboratory, Toshiba Corporation, 4-1, Ukishima-cho, Kawasaki-ku, Kawasaki, Kanagawa 210 Japan, e-mail: nobuatsu.tanaka@toshiba.co.jp

Although the linear system can be is solved using a suitable iterative matrix solver, the convergence speed deteriorates with an increase in the coefficient matrix condition number $\kappa(A)$, where

$$\kappa(A) = ||A|| \cdot ||A^{-1}||$$

for a specified matrix norm $||\cdot||$. The condition number increases suddenly with an increase in the number of mesh points.

Two approaches can be used to repress the marked increase in computing time: the computational algorithm can be improved (e.g., by reducing the number of iterations) or more powerful hardware can be employed (e.g., powerful vector or parallel supercomputers). We use both approaches in our study to obtain significantly better results than have been found in previous works; a new algorithm is developed to reduce computational costs, and the algorithm is implemented on a high-performance computer. Our algorithm suppresses the marked increase of the computing time by applying a wavelet-based preconditioner to the coefficient matrix. This approach follows from an observation that the condition number can be bounded within a limited value range by using wavelet-based techniques [1] [5].

Wavelets are an important new mathematical tool for describing complex functions and analyzing empirical continuous data derived from many different types of signals. In 1988 Daubechies [2] proposed families of compactly supported wavelets with user-specifiable degrees of smoothness, constructed from a multiresolution analysis (MRA). Studies by researchers using these wavelets for solving partial differential equations include [1], [5]. In one study Beylkin[1] proposes a new algorithm to solve a linear system using the discrete wavelet transform (DWT) which prevents and increase in the condition number of a large matrix. We modify his method – a method which is difficult to implement to solve practical problems, because it assumes periodic boundary conditions (BCs). Our technique, the incomplete discrete wavelet transform (iDWT) [7], [10], [9] does not assume periodic BCs. Instead, it assumes that all values outside the boundaries are equal to zero. Our method suppresses the increase in the condition number of a large matrix through diagonal re-scaling. The effects and advantages of the iDWT preconditioning are confirmed with one-dimensional boundary value problems (BVPs) of elliptic equations. Extensions to multi-dimensional problems are also investigated. The remainder of this paper is organized as follows. In the next section, we begin with a brief review of wavelets, then one- and two-dimensional iDWT algorithms are given. The third section describes our implementation studies using the Cray T3D parallel supercomputers. The fourth and concluding section is a summary of our findings.

2 The Incomplete Discrete Wavelet Transform

In this section we review basic concepts associated with scaling functions and wavelets. Basis sets for subspaces of $L^2(R)$ are described. We select scaling functions and wavelets to be the basis sets and review how they can be used to approximate functions in $L^2(R)$. The main portion of this section consists of a presentation of our one- and two-dimensional incomplete discrete wavelet transform (iDWT) algorithm.

2.1 One-dimensional problem

Consider the one-dimensional elliptic equation

$$-\Delta u = f \quad \text{in} \quad \Omega = [0,1], \tag{1}$$

$$u = g \quad \text{on} \quad \partial\Omega \quad \text{or} \quad \frac{\partial u}{\partial x} = 0 \quad \text{on} \quad \partial\Omega.$$

We discretize this problem on a uniform mesh to obtain a system of linear equations of the form

$$Au = f, \tag{2}$$

where $u = \{u_i\}, f = \Delta x^2\{f_i\}$, the coefficient matrix A is a 2^n-by-2^n, tri-diagonal, positive-definite matrix of the form

$$A = \underbrace{\left.\begin{pmatrix} a & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & & \vdots \\ 0 & -1 & 2 & \ddots & 0 \\ \vdots & & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & b \end{pmatrix}\right\} 2^n}_{2^n},$$

and a and b are real numbers which represent the BCs at each edge point. Hereafter, we assume $N = 2^n$. Before explaining the iDWT, we briefly review the wavelets and set our notation. In this study, we use the Daubechies wavelet with M vanishing moments (M is positive integer). Let us define $\varphi_{j,k}$ and $\psi_{j,k}$ as the Daubechies scaling function and wavelet function of j-th order resolution. These functions are formed by the dilation and translation of basic functions φ and ψ,

$$\varphi_{j,k}(x) = 2^{(n-j)/2}\varphi\left(2^{n-j}x - k\right) \quad , \quad \psi_{j,k}(x) = 2^{(n-j)/2}\psi\left(2^{n-j}x - k\right).$$

By defining V_j as a subspace of $L^2(\Omega)$ and O_j as an orthogonal complement of V_j in V_{j-1}, then the space V_{j-1} is represented as a direct sum

$$V_{j-1} = V_j \oplus O_j.$$

Because $\varphi_{j,k}$ is an orthonormal basis of V_j and $\psi_{j,k}$ is an orthonormal basis of O_j for every j, the following relations are held.

$$\varphi_{j,k}(x) = \sum_i h_k \varphi_{j-1,2k+i}(x) \quad , \quad \psi_{j,k}(x) = \sum_i g_k \varphi_{j-1,2k+i}(x),$$

where $\{h_i\}, \{g_i\}$ are the wavelet filter coefficients unique for a wavelet base [2]. The Daubechies' wavelet coefficients satisfy

$$g_i = (-1)^i h_{1-i}.$$

A function $u(x) \in L^2(\Omega)$, can be approximated by a scaling function and wavelet expansion

$$u(x) \cong \sum_k s_k^m \varphi_{m,k}(x) + \sum_{j=1}^{m}\sum_k d_k^j \psi_{j,k}(x) \ , \tag{3}$$

where s_k^j and d_k^j are the coefficients of the expansion

$$s_k^j = \int_\Omega u(x)\,\varphi_{j,k}(x)\,dx \ , \ d_k^j = \int_\Omega u(x)\,\psi_{j,k}(x)\,dx \ .$$

When n is sufficiently large and $u(x)$ sufficiently smooth, we can use an even simpler approximation of the coefficients of basic resolution level ($j = 0$)

$$s_k^0 \cong 2^{-n/2} u_k$$

[10]. This simpler expression allows us to generate the coefficients $\{s_k^0\}$ using only the data $\{u_i\}$ and $\{f_i\}$ $(\times 2^{-n/2})$ which are defined on the grid nodes. The coefficients in equation (3) are evaluated using the pyramid algorithm given by the equations

$$s_k^j = \sum_i h_i s_{2k+i}^{j-1} \ , \ d_k^j = \sum_i g_i s_{2k+i}^{j-1}.$$

Since the above described wavelet-based transformations are orthonormal, we refer to them method as the complete discrete wavelet transform (cDWT). Straightforward implementation of the pyramid algorithm leads to difficulties in handling boundary points since the procedure requires several data values which are defined outside the boundaries and are not known (see Figure 1). To overcome this difficulty, Beylkin [1] introduced a periodic BC requirement. However, this requirement introduces a new difficulty, namely, a means for transforming general BVPs to ones with periodic BCs. Short of a simple answer, the method is too complicated to adopt for solving general problems.

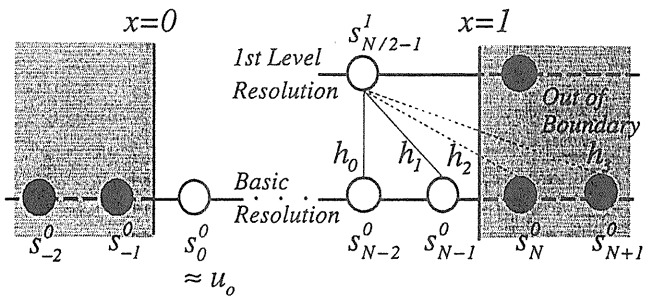

Figure 1: The pyramid algorithm and boundary points

Our incomplete discrete wavelet transforms (iDWT) method overcomes the problems associated with the cDWT method, because it does not require periodic BCs and assumes that all values outside the boundaries are equal to zero. Although the transformations are not orthonormal, the algorithm is more suitable and effective than its complete counterpart for numerical simulations and is much simpler to implement.

Let $W_{(j)}$ denote the iDWT matrix which transforms coefficient data from the basic to the j-th order resolution. Then

$$W_{(j)} = W_{j-1} W_{j-2} \cdots W_0 \ ,$$

where W_j is the $N \times N$ matrix which transforms data from the $(j-1)$-th to the j-th resolution, i.e.,

$$W_j = \begin{pmatrix} \begin{pmatrix} H_j \\ G_j \end{pmatrix} & 0 \\ 0 & I_{N-2^{n-j}} \end{pmatrix} \ ,$$

I_i is the $i \times i$ unit matrix, and H_j and G_j are 2^{n-j}-by-2^{n-j-1} banded matrices defined only by the wavelet filter coefficients [10]. Equation (2) can be expressed in terms of these matrices as

$$W_{(j)} A W_{(j)}^{-1} \tilde{u}_j = \tilde{f}_j \ ,$$

where

$$\tilde{u}_j = W_{(j)}u \ , \ \tilde{f}_j = W_{(j)}f \ .$$

Both $\tilde{u}_j$ and $\tilde{f}_j$ are composed of wavelet expansion coefficients of $u(x)$ and $f(x)$. Since $W_{(j)}$ approximates the cDWT which is an orthonormal transform,

$$W_{(j)}W_{(j)}^T \cong I_N \ ,$$

so that

$$W_{(j)}AW_{(j)}^T\tilde{u}_j \cong \tilde{f}_j \ . \tag{4}$$

2.2 Diagonal Rescaling Method

When the linear system (2) is transformed to a space with wavelets as basis functions, the matrix A which represents the Laplacian operator is transformed to a matrix $W_{(j)}AW_{(j)}^T$. When the functions are periodic in Ω, the diagonal rescaling matrix [1],[10] takes the form

$$P_j = \mathrm{diag}\,(\,2^jI_{2^{n-j+1}} \quad 2^{j-1}I_{2^{n-j+1}} \quad 2^{j-2}I_{2^{n-j+2}} \quad \cdots \quad 2I_{2^{n-1}}\,)$$

for $j > 1$, and

$$P_j = 2I_{2^n}$$

for $j = 1$. This rescaling matrix suppresses an increase in the condition number of the matrix when the Conjugate Gradient (CG) iterative method is used to solve a linear system. We apply the matrix P_j to system (4) to obtain

$$P_jW_{(j)}AW_{(j)}^TP_j\bar{u}_j \ \cong \ \bar{f}_j \ , \tag{5}$$

where

$$\bar{u}_j = P_j^{-1}W_{(j)}u \ , \ \bar{f}_j = P_jW_{(j)}f \ .$$

System (5), which has been diagonally re-scaled, can now be solved numerically using the CG method.

2.3 iDWT PCG method

We apply the iDWT to the Preconditioned Conjugate Gradient (PCG) Method by transforming (2) to the form

$$P_jW_{(j)}AW_{(j)}^TP_j\hat{u}_j = \hat{f}_j \ , \tag{6}$$

where

$$\hat{u}_j = 2^{-n/2}P_j^{-1}W_{(j)}^{-T}u \ , \ \hat{f}_j = 2^{-n/2}P_jW_{(j)}f \ .$$

Note that while equation (5) approximates (2), equation (6) is equivalent to (2). To solve (6) using the PCG method, set $V = P_jW_{(j)}$, and the preconditioning matrix K to

$$K = V^TV = W_{(j)}^TP_j^2W_{(j)} \ .$$

For a one-dimensional Poisson equation solver, the operation P_j^2 can be simplified to the change in filter coefficients

$$\{h'_k\} = 2\,\{h_k\} \quad \text{and} \quad \{g'_k\} = 2\,\{g_k\}$$

[7]. We compute the preconditioned residual

$$\tilde{r} = W_{(j)}^TP_j^2W_{(j)}r$$

for the the PCG algorithm using this change in filter coefficients. The system is then solved using the PCG method.

2.4 Two-dimensional problem

Consider the two-dimensional Poisson equation,

$$\begin{aligned} -\Delta u &= f \quad \text{in} \quad \Omega = [0,1] \times [0,1] \ , \\ u &= g \quad \text{on} \quad \partial\Omega \ . \end{aligned} \tag{7}$$

To solve this problem numerically, we use a finite difference scheme with a uniform mesh on region Ω and the direct product of two one-dimensional orthonormal bases to extend the iDWT to two dimensions, i.e.,

$$V_{j-1}^{(xy)} = \left(V_j^{(x)} \otimes V_j^{(y)}\right) \oplus \left(V_j^{(x)} \otimes W_j^{(y)}\right) \oplus \left(W_j^{(x)} \otimes V_j^{(y)}\right) \oplus \left(W_j^{(x)} \otimes W_j^{(y)}\right) \ .$$

Then, the scaling function and wavelet expansion of $u(x,y) \in L^2(\Omega)$ is

$$u\,(x,y) \cong \sum_{l,m} s_{l,m}^L \varphi_l^L \varphi_m^L + \sum_{j=1}^{L} \sum_{l,m} \left\{ d_{l,m}^{(x),j} \psi_l^j \varphi_m^j + d_{l,m}^{(y),j} \varphi_l^j \psi_m^j + d_{l,m}^{(xy),j} \psi_l^j \psi_m^j \right\} ,$$

with the expansion coefficients

$$\begin{aligned} s_{l,m}^j &= \sum_{i,k} h_i h_k s_{2l+i,2m+k}^{j-1} \ , \quad d_{l,m}^{(x),j} = \sum_{i,k} g_i h_k s_{2l+i,2m+k}^{j-1} \ , \\ d_{l,m}^{(y),j} &= \sum_{i,k} h_i g_k s_{2l+i,2m+k}^{j-1} \ , \quad d_{l,m}^{(xy),j} = \sum_{i,k} g_i g_k s_{2l+i,2m+k}^{j-1} \ . \end{aligned}$$

As in the one-dimensional case, diagonal rescaling for the two-dimensional Poisson equation amounts to transformation of filter coefficients in each coordinate [10].

$$\{h_i'\} = \sqrt{2}\,\{h_i\} \quad \text{and} \quad \{g_i'\} = \sqrt{2}\,\{g_i\} \ .$$

3 Numerical Examples

Our numerical implementations of the iDWT algorithm for one- and two-dimensional problems are described in this section. We compare our results with those from use of standard techniques, such as the conjugate gradient (CG), successive over-relaxation (SOR) and incomplete Cholesky conjugate gradient (ICCG [6]) methods. Implementation studies using Cray T3D parallel supercomputers are also described.

3.1 One-dimensional problem

We consider the numerical solution of the one-dimensional Poisson equation. Table 1 shows our results in which a uniform mesh of N grid points were used to solve the Dirichlet BVP, with and without iDWT preconditioning, using Daubechies orthonormal wavelets with 3 vanishing moments [2].

Table 1: $\kappa(A)$ for the Dirichlet BVP with & without iDWT preconditioning
$(a = b = 2, M = 3, j = n - 2)$

grid points	A	$P_j W_{(j)} A W_{(j)}^T P_j$
32	440.7	27.0
64	1713	53.7
128	6742	107.6
256	26800	216.0

The results show that iDWT preconditioning method effectively suppresses the increase in the condition number when the number of grid points is increased. From the eigenvalue analysis [10], the reason was found to be that while eigenvalues of the original matrix are uniformly distributed throughout, iDWT preconditioning moves most of the eigenvalues close together, towards the median range. This new distribution prevents an increase in the condition number as the number of grid points increases and facilitates the use of CG algorithms by reducing the number of iterations required for convergence and the associated computing time.

We also investigated the effect of the iDWT on several kinds of BVPs including Neumann and the mixed BVPs. From the results using Daubechies' wavelet with 3vanishing moments, preconditioning in both BVPs suppresses the increase in the condition number when the number of grid points is increased [10].

3.2 Two-dimensional problem

3.2.1 Basic characteristics of major iterative solvers

Our studies described above show the effectiveness of iDWT preconditioning for solving the one-dimensional Poisson equation. We now consider the two-dimensional problem since two or more dimensions are normally required for modeling real fluid dynamics problems. We begin by analyzing the theoretical performance of two basic solvers, the Conjugate Gradient (CG) and the Successive Over Relaxation (SOR) methods for solving problem (7).

First, we consider the CG method. If we assume the convergence criterion ε

$$\|u - u_k\|_A \leq \varepsilon \|u - u_0\|_A ,$$

then the number of iterations required for convergence of the CG method without preconditioning satisfies

$$k \leq -\frac{\log \varepsilon}{\pi}(N+1) ,$$

so that the number is proportional to N.

Effective use of SOR solvers depends on selecting the optimal (or near optimal) acceleration parameter for a given problem. In the optimal parameter case for problem (7), we make an approximation of the asymptotic convergence ratio [11] with

$$R_\infty = -\ln \rho_{opt} \cong \frac{\pi}{N+1}$$

[10] and conclude that the number of iterations required for convergence is proportional to the number of grid points N when the optimal acceleration parameter is used. In the general case for a fixed acceleration parameter $\omega < \omega_{opt}$, the asymptotic convergence ratio becomes

$$R_\infty = -\ln \rho_{const} \cong \frac{\omega}{2-\omega} \cdot \frac{\pi^2}{(N+1)^2}$$

[10]. The number of iterations required for convergence increases proportionately to N^2, i.e., the speed of convergence is worse than the optimal parameter case.

Table 2: Iterations required to solve 2-D problem

N	SOR ($\omega = \omega_{opt}$)	SOR ($\omega = 1.9$)	CG	ICCG (1,1)	iDWTCG (M=3 , j=n-2)
128	543	1643	220	102	59
256	1110	6514	452	175	77
512	2270	25119	931	361	102
1024	not implemented		1910	709	131

We compare our theoretical results with numerical implementations with the convergence criterion

$$\|r_i\|_2 = \|Au_i - f\|_2 \leq 1.0 \times 10^{-8}\|f\|_2.$$

Our results (see Table 2) show that the number of iterations required for the CG and SOR methods using the optimal acceleration parameter are proportional to N and for SOR with a fixed acceleration parameter N^2. They are consistent with our theoretical analysis described above. Furthermore, we observed that the iDWTCG method significantly lowers the rate of increase in the number of iterations required for convergence when the number of grid points increases.

3.2.2 Parallel Processing

As the vector processing results of iDWTCG are detailed in [7], [9], [10], we pay attention to the parallel processing in this section. We determined a numerical solution of (2) using the iDWTCG on a Cray T3D, a parallel MIMD with distributed memory in which processor intercommunication is facilitated by 3-dimensional torus networking. The work-sharing model, the simplest of three commonly used types of parallel programming models (the other two being procedure decomposition and data decomposition) was used in our studies. The reason why we may use the simplest model is that DWT algorithm originally consists of simple local data processing and is fitted for the parallel processing. However, the ordinary data arrangement is not suitable for parallel processing since it leads to a great deal of processor idling and data communication. We modify the data arrangement to to reduce the number of idle processors by more evenly balancing computational loads and reducing data communication overhead [10].

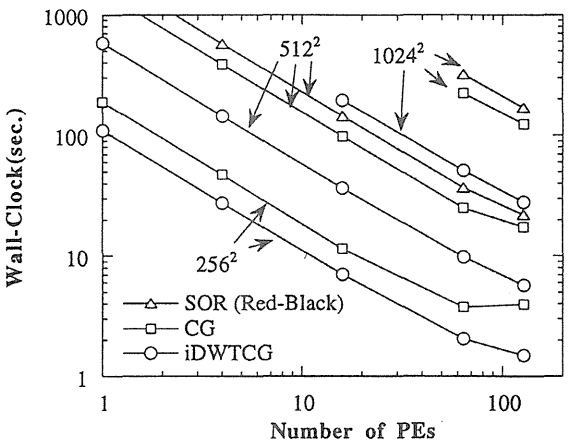

Figure 2: Parallel performance of modified iDWTCG solver

Results from the implementations using Daubechies' wavelets with 3 vanishing moments with 512×512 grid points are shown in Figure 2. We make it possible for SOR to process in parallel by reducing data recurrence relations with Red-Black ordering [3]. The figure shows that the computing time of the iDWTCG method decreases at the same rate as the other methods. The detailed results including parallel scalability and efficiency are described in [10]. From the results, it is worth noting that solving 1024×1024 size problem with iDWTCG takes only the same seconds of solving 512×512 problem with SOR or CG.

4 Conclusion

In this paper, we proposed a new preconditioning method called the the incomplete discrete wavelet transform (iDWT), then applied the method to solve Poisson equations. iDWT preconditioning improves the

condition number of linear systems by shifting the eigenvalues of the coefficient matrix towards medium range values. The method is highly suitable use prior to the application of iterative, projection type matrix solvers, such as the CG method. The performance of the iDWTCG method becomes increasingly better than that of the ICCG method as the size of of a problem becomes larger (for example, when successively finer meshes are used). And we demonstrated that our preconditioning scheme is equally effective for the solution of multi-dimensional problems.

Since our iDWTCG algorithm consists soley of local operations, it achieves high computing performance on vector and parallel computers. For example, on a Cray C94D vector computing system, the iDWTCG method can solve 2-dimensional discretized Poisson equation's using 1024×1024 grid points, about 14 times faster than the ICCG method. Our iDWTCG method is highly efficient preconditioning method for CG when it the code is tuned to run on parallel processing machines, because it contains much fewer data recurrence relations than the IC preconditioning. (In fact, IC preconditioning contains so many data recurrence relations that its efficiency drops to a level which is worse than a system without preconditioning [8]). The iDWTCG also leads to excellent performance on a Cray T3D parallel computer; speed up by a factor of 100 is achieved by using 128 processors.

References

[1] G. Beylkin (1993), "On wavelet-based algorithms for solving differential equations", pp. 449-466 in *Wavelets: Mathematics and Applications*, J. Benedetto, M. Frazier (eds.), CRC Press, Inc., Tokyo.

[2] I. Daubechies (1988), "Orthonormal bases of compactly supported wavelets", *Comm. Pure Appl. Math.*, vol. 41, pp. 909–996.

[3] I. Duff and G. Meurant (1989), "The effect of ordering on preconditioned conjugate gradients", *BIT* 29, pp. 635 – 657.

[4] M. Hestenes and E. Steifel (1962), "Methods of conjugate gradient for solving linear systems", *J. Res. Nat'l. Bur. Stds.*, vol. 49, pp. 409–436.

[5] S. Jaffard and Ph. Laurençot (1992), "Orthonormal Wavelets, Analysis of Operators and Applications to Numerical Analysis", pp. 543–601 in *Wavelets: A Tutorial and Applications*, C.K. Chui (ed.), Academic Press, Boston.

[6] J. Meijerink and H. van der Vorst (1977), "An iterative solution method for linear systems for which the coefficient matrix is a symmetric M-matrix", *Math. Comp.*, vol. 31, pp. 148–162.

[7] N. Tanaka et al. (1996), "Incomplete Discrete Wavelet Transform and the Application to a Poisson Equation Solver," J. Nucl. Sci. Technol., vol. 33, no. 7, pp. 555–561.

[8] N. Tanaka and H. Terasaka (1993), "A Numerical Study of Incompressible Viscous Flow with Massively Parallel Processors," Comp. Fluid Dynamics J., vol. 2, no. 2, pp. 145–160.

[9] N. Tanaka (1997), "Application of Incomplete Discrete Wavelet Transform to a Poisson Equation Solver and Its Parallel Processing", *Trans. Jpn. Soc. Indust. Appl. Math.*, vol. 7, no. 4, to be appeared (in Japanese).

[10] N. Tanaka, "A Wavelet-based Conjugate Gradient Method for Solving Poisson Equations", *Wavelets and Their Applications: Case Studies*, M. Kobayashi (ed.), submitted to SIAM for review.

[11] R. Varga (1962), *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ.